Caching

micro-23

Aliaksei Bialiauski

Designed in LATEX

All visual and text materials presented in this slidedeck are either originally made by the author or taken from public Internet sources, such as website. Copyright belongs to their respected authors.

Rationale of Caching

Eviction policies

Caching patterns

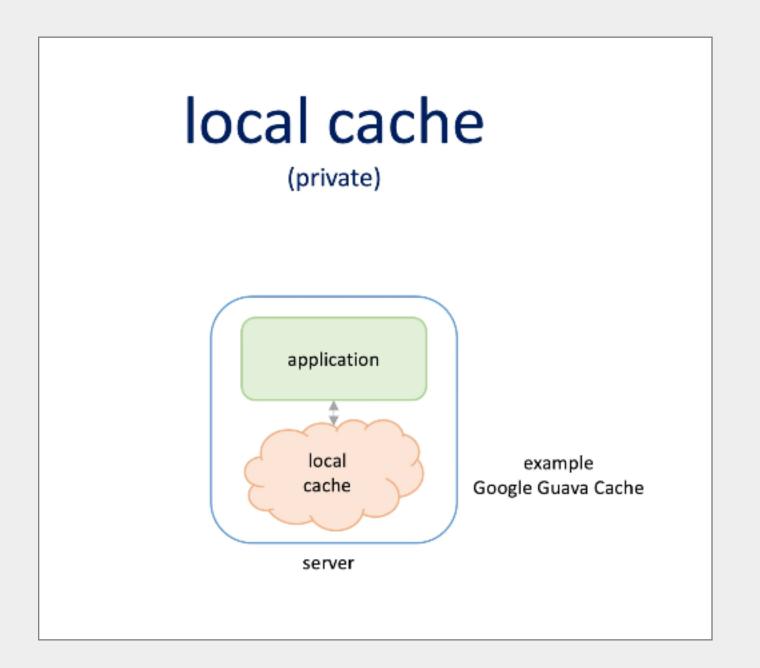
Rationale Eviction Patterns 3/16

Chapter #1:

Rationale of Caching

Rationale Eviction Patterns

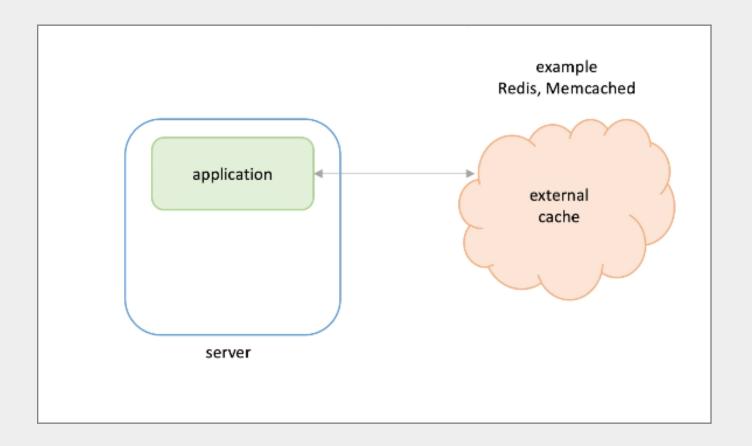
Local Cache



Caching

Rationale Eviction Patterns 5/16

External Cache



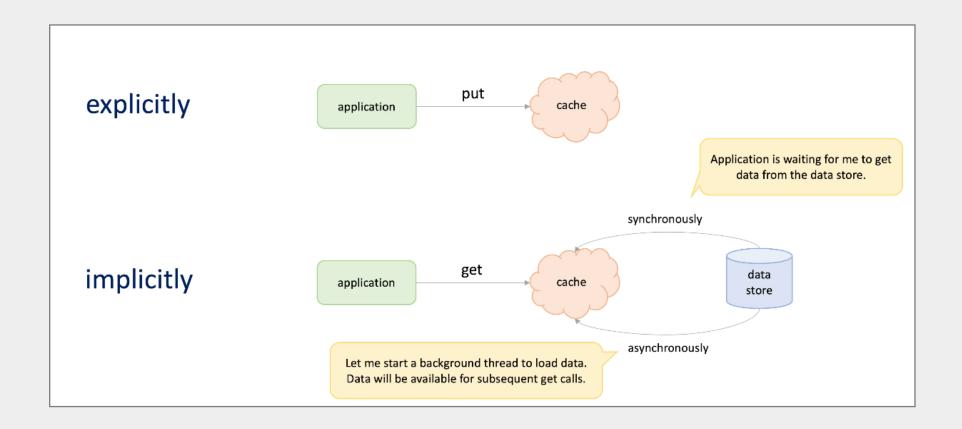
Rationale Eviction Patterns 6/16

Chapter #2:

Eviction policies

Rationale Eviction Patterns 7/16

How do we add data to cache?

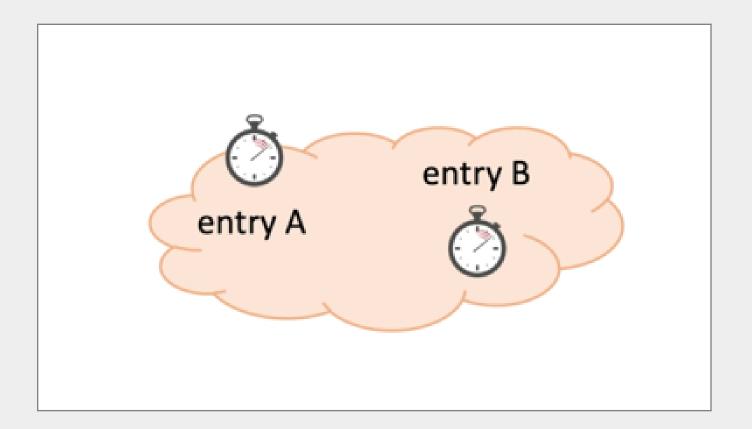


LRU vs. LFU

"Least Recent Used (LRU) means evict entries that haven't been used recently."

"Least Frequently Used (LFU) means evic entries that were used least often."





Rationale Eviction Patterns 10/16

Active vs. Passive expiration

"Active - when entry is accessed."

"Passive - Background thread that runs at regular intervals."

Rationale Eviction Patterns 11/16

Explicitly remove-based eviction

```
class ExplicitEviction {
  private final Cache cache;
  private final Client client;

  public static void main(String[] args) {
    this.cache.put(
      1L,
      new Book("Martin Fowler", "Refactoring", "0134757599")
    );
    this.client.book(1L); // from cache
    this.cache.evict(1L);
    this.client.book(1L); // from db
  }
}
```

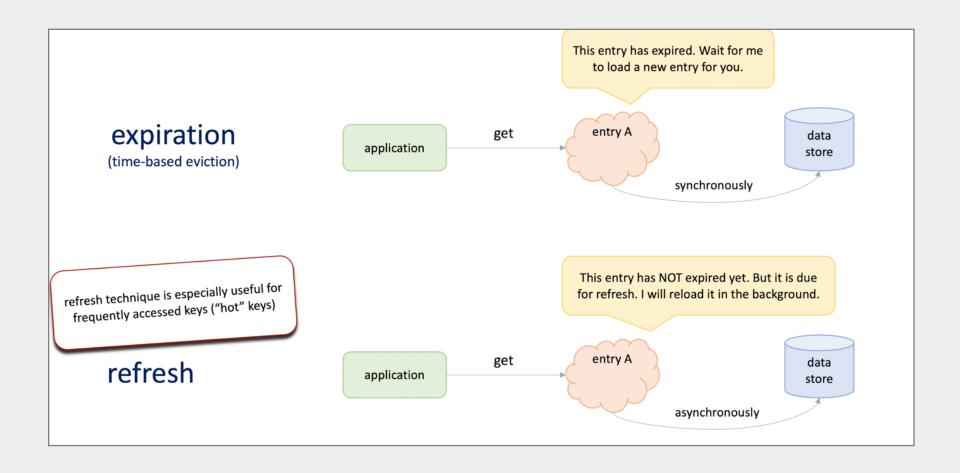
Caching Ching Chin

Rationale Eviction Patterns 12/16

Chapter #3:

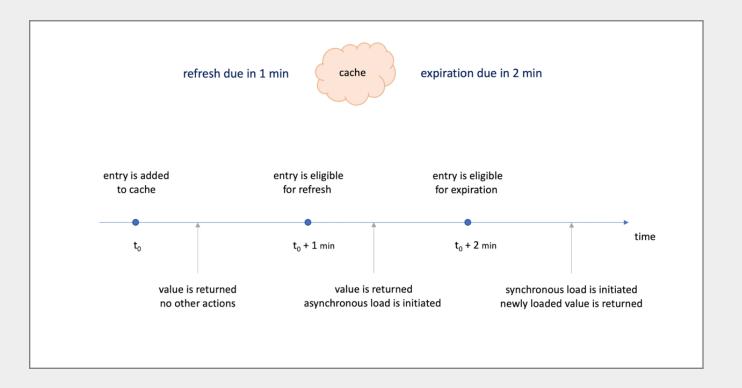
Caching patterns

Expiration vs. Refresh

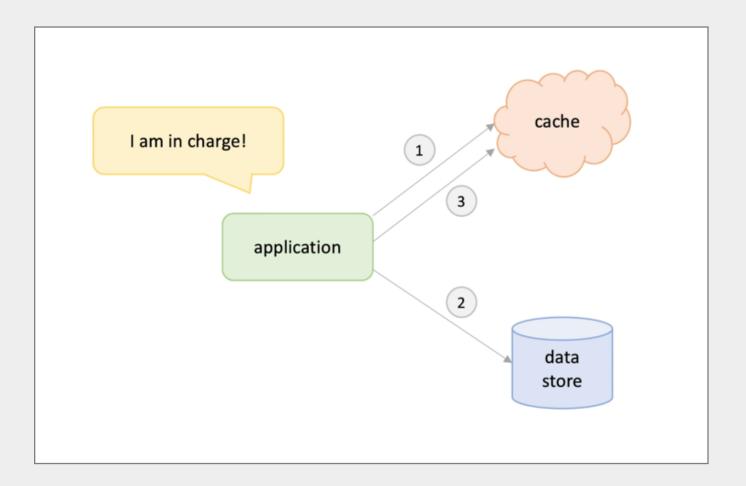


Rationale Eviction Patterns 14/16

Refresh-ahead



Cache-aside



Caching Ching Chin

Rationale Eviction Patterns 16/16

Read/Write through Cache

